

MULTIMEDIA



UNIVERSITY

STUDENT ID NO

--	--	--	--	--	--	--	--	--	--

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2016/2017

TPL 2141 – PROGRAMMING LANGUAGE CONCEPT

(All sections / Groups)

19 OCT 2016
2.30 p.m – 4.30 p.m
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This Question paper consists of 4 pages (excluding cover page) with 5 Questions.
2. Attempt **ALL** questions. All questions carry equal marks and the distribution of the marks for each question is given.
3. Please write all your answers in the Answer Booklet provided

Question 1

(a) Given two sample codes of different categories of programming language.

```
#include <iostream>
```

```
int main()
```

```
{
```

```
    std::cout << "Hello World!"
```

```
        << std::endl;
```

```
    return 0;
```

```
}
```

Language 1

```
<html>
```

```
<header>
```

```
<title>This is title</title>
```

```
</header>
```

```
<body>
```

```
    Hello world
```

```
</body>
```

```
</html>
```

Language 2

(i) High-level programming language can be implemented with compilation, pure interpreted or hybrid implementation. Discuss (with diagram illustration) the implementation method for sample code in Language 1 and 2.

(ii) Language 1 is claimed to be more reliable but less readable than Language 2. Describe the characteristic of the language in terms of readability and reliability.

[3 + 2 = 5 marks]

(b) In compilation process, lexical and syntax analyser are responsible to translate source language and pass on to intermediate code generator.

(i) Specify the primary task of lexical analyser.

(ii) Why is state transition diagram useful in lexical analyser? Give an example of state diagram that recognize an IDENTIFIER token.

[1.5 + 3.5 = 5 marks]

Continued

Question 2

Consider the following grammar with start symbol (Declaration) and terminals: int, char, *, number, identifier, '[', ']', '(', ')':

Declaration $\rightarrow$ Type Declarator
 Type $\rightarrow$ int | char
 Declarator $\rightarrow$ * Declarator | Declarator '[' number ']' |
 Declarator '(' Declaration ')' | '(' Declarator ')' | identifier

- (a) Draw parse tree for the sentence: char *name[5] [3 marks]
- (b) Show the rightmost derivation (without using parse tree) for sentence (a). [3 marks]
- (c) Is this grammar ambiguous? Prove with this sentence: int count (int step) [4 marks]

Question 3

- (a) Some languages distinguish between uppercase and lowercase in identifiers.
- (i) What are the advantage and disadvantage of the case-sensitive design? Explain.
- (ii) Is C++ implementing the case-sensitive design? Prove with the example of codes. [2 + 2 = 4 marks]
- (b) Given the rules of operator precedence and associativity for arithmetic expression:

Precedence	Highest	*, / +, - - (unary)
Associativity	Lowest Left to Right	=, /, <, <=, >=, >

- (i) How do you apply operator precedence and associativity in arithmetic expression? Prove with an example.
- (ii) Short-circuit evaluation happens in the expression of some programming languages, where (b/13-1) is ignored when a is zero. Why?
 (13 * a) * (b / 13 -1)

[3 + 3 = 6 marks]

Continued

Question 4

- (a) `for` statement is an iterative control statement for procedural language.
- (i) How many types of expression statement are there in a `for` statement? How many times each type of expression is evaluated in a `for` statement?
 - (ii) Discuss the difference of `for` statement in C++ and Java.
- [3 + 1 = 4 marks]
- (b) Each variable is an abstraction of a memory cell and categorised into 4 types of lifetime: static, stack-dynamic, explicit heap-dynamic and implicit heap dynamic. Compare any **TWO** of the variable lifetime.
- [2 marks]
- (c) Subprogram is an abstraction method to specify complex operations and is only executed when it is called by a calling program. When do you need to use subprogram declaration and subprogram definition? Explain with an example.
- [4 marks]

Question 5

- (a) Analyse the following Java code:

```
outerLoop:
    for (row = 0; row < numRows; row++)
        for (col = 0; col < numCols; col++) {
            sum += mat[row][col];
            if (sum > 1000.0)
                break outerLoop;
        }
```

- (i) Find the unconditional labeled exit and explain the purpose of using unconditional **labeled** exit.
- (ii) C, C++, Python, and C# only have unconditional **unlabeled** exit. Show an example of code with unconditional **unlabeled** exit.

[2 + 2 = 4 marks]

Continued

(b) Analyse the following pseudocode:

```
z : integer // global variable

subprogram p (x:integer)
{
    x := x+1
    z := z+2
}

calling program:
z := 1
p(z)
write(z)
```

- (i) Find out the actual and formal parameters of subprogram.
- (ii) Compute the value of z based on passed by value, passed by reference and passed by value-result procedures.

[1 + 3 = 4 marks]

(c) Analyse the following pseudocode and calling sequence:

```
functionbig() {
    function sub1()
        var x = 7
    function sub2() {
        var y = x
    }
    var x = 3
}
```

Calling Sequence:

```
big calls sub1
sub1 calls sub2
sub2 uses x
```

Explain the process of finding the output of x by assuming the cases: dynamic and static scoping.

[2 marks]

End of Page